

## The Claims

1. (Currently amended) A software architecture for a distributed computing system implemented at least in part by a computing device, the software architecture comprising:

an application configured to handle requests submitted by remote devices over a network; and

an application program interface to present functions used by the application to access network and computing resources of the distributed computing system, wherein the application program interface comprises a set of classes that make available standards-based support for processing XML documents, wherein the set of classes are grouped in the application program interface into a plurality of namespaces, and wherein a first of the plurality of namespaces contains classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) and a second of the plurality of namespaces contains an XPath parser and evaluation engine, wherein the set of classes comprises:

an XmlReader class that enables non-cached forward only access to XML data;

an XPathNavigator class that enables read-only random access to a data store;

an XslTransform class that enables transforming of XML data using an XSLT stylesheet;

a plurality of XML Schema classes that enable constructing and editing of schemas;

an XmlResolver class that enables resolving of external XML resources named by a Uniform Resource Identifier (URI);

an XmlDocument class that enables structured data to be stored, retrieved, and manipulated through a relational dataset; and

an XmlWriter class that enables a non-cached forward only way of generating streams and files containing XML data.

2. (Canceled).

3. (Currently amended) A software architecture as recited in claim 1[[2]], wherein the set of classes further comprises:

an XmlValidatingReader class that enables DTD, XDR and XSD schema validation.

4. (Original) An XmlReader class of an application program interface, embodied on one or more computer readable media that enables non-cached forward only access to XML data, the XmlReader class comprising:

an XmlReader constructor that enables initialization of a new instance of the XmlReader class; and

a Read method that enables reading of nodes of the XML data via the XmlReader class instance.

5. (Original) An XmlReader class of an application program interface as recited in claim 4, wherein the XmlReader class further comprises:

a BaseURI property that identifies a base URI of a current node of the XML data; and

a NodeType property that identifies the type of the current node.

6-8. (Canceled).

9. (Original) An XmlResolver class of an application program interface, embodied on one or more computer readable media, that enables resolving of external XML resources named by a Uniform Resource Identifier (URI), the XmlResolver class comprising:

a ResolveURI method that enables resolving the absolute URI from a base URI and a relative URI; and

a GetEntity method that enables mapping of the resolved URI to an object containing identified resource.

10. (Original) An XmlDataDocument class of an application program interface, embodied on one or more computer readable media, that enables structured data to be stored, retrieved, and manipulated through a relational dataset, the XmlDataDocument class comprising:

a DataSet property that enables obtaining of a dataset that provides a relational representation of the data in a document;

a Load method that enables loading of the document using a specified data source and synchronizing the dataset with the loaded data.

11-15. (Canceled).

16. (Currently amended) A distributed computer software architecture implemented at least in part by a computing device, comprising:

one or more applications configured to be executed on one or more computing devices, the applications handling requests submitted from remote computing devices;

a networking platform to support the one or more applications; and

an application programming interface to interface the one or more applications with the networking platform, wherein the application program interface comprises a set of classes that make available standards-based support for processing documents written in a markup language, wherein the set of classes are grouped in the application programming interface into a plurality of namespaces, and wherein a first of the plurality of namespaces contains classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations), a second of the plurality of namespaces contains an XPath parser and evaluation engine, and a third of the plurality of namespaces contains classes used to serialize objects into XML format documents or streams, wherein the set of classes comprises:

an XmlReader class that enables non-cached forward only access to XML data;

an XPathNavigator class that enables read-only random access to a data store;

an XslTransform class that enables transforming of XML data using an XSLT stylesheet;

a plurality of Xml Schema classes that enable constructing and editing of schemas;

an XmlResolver class that enables resolving of external XML resources named by a Uniform Resource Identifier (URI);

an XmlDataDocument class that enables structured data to be stored, retrieved, and manipulated through a relational dataset; and

an XmlWriter class that enables a non-cached forward only way of generating streams and files containing XML data.

17. (Canceled).

18. (Currently amended) A software architecture as recited in claim 16[[17]], wherein the set of classes further comprises:

an XmlValidatingReader class that enables DTD, XDR and XSD schema validation.

19-21. (Canceled).

22. (Currently amended) A method comprising:  
receiving one or more calls from one or more remote devices over a network, wherein the one or more calls are to one or more functions that make available support for processing XML documents, the one or more functions being grouped into a plurality of namespaces with a first namespace containing an XPath parser and evaluation engine and a second namespace containing classes used to serialize objects into XML format documents or streams, wherein the one or more functions comprises:

an XmlReader function that enables non-cached forward only access to XML data;

an XPathNavigator function that enables read-only random access to a data store;

an XslTransform function that enables transforming of XML data using an XSLT stylesheet;

a plurality of Xml Schema functions that enable constructing and editing of schemas;

an XmlResolver function that enables resolving of external XML resources named by a Uniform Resource Identifier (URI);

an XmlDataDocument function that enables structured data to be stored, retrieved, and manipulated through a relational dataset; and

an XmlWriter function that enables a non-cached forward only way of generating streams and files containing XML data; and  
performing the requested XML document processing.

23. (Cancelled).
24. (Currently amended) A ~~software-architecture method~~ as recited in claim 22[[23]], wherein the one or more functions further comprises:  
an XmlValidatingReader function that enables DTD, XDR and XSD schema validation.
25. (Currently amended) A method comprising:  
calling, to one or more remote devices over a network, one or more functions that make available support for processing XML documents, the one or more functions being grouped into a plurality of namespaces with a first namespace containing classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) and a second namespace containing classes used to serialize objects into XML format documents or streams, wherein the one or more functions comprises:  
an XmlReader function that enables non-cached forward only access to XML data;  
an XPathNavigator function that enables read-only random access to a data store;  
an XslTransform function that enables transforming of XML data using an XSLT stylesheet;  
a plurality of Xml Schema functions that enable constructing and editing of schemas;

an XmlResolver function that enables resolving of external XML resources named by a Uniform Resource Identifier (URI);

an XmlDataDocument function that enables structured data to be stored, retrieved, and manipulated through a relational dataset; and

an XmlWriter function that enables a non-cached forward only way of generating streams and files containing XML data;

receiving, from the one or more remote devices, a response to the calling.

26. (Cancelled).

27. (Currently amended) A software-architecture method as recited in claim 25[[26]], wherein the one or more functions further comprises:

an XmlValidatingReader function that enables DTD, XDR and XSD schema validation.